

IN THE CLAIMS:

This listing of the claims replaces all prior versions and listings of the claims in this application.

The text of all pending claims (including any withdrawn claims) is set forth below. Canceled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~striketrough~~. The status of each claim is indicated with one of (Original), (Currently amended), (Canceled), (Withdrawn), (Previously presented), (New), and (Not entered).

Please AMEND claims 1, 3, 5, 6, 14, 15, 17, 18, 23-25, 28, 30, 32, 33, 35, and 36, CANCEL claims 19, 34, 37, and 38 without prejudice or disclaimer, and ADD new claims 39-44 in accordance with the following:

1. (Currently amended) A non-transitory computer-readable storage medium usable with an apparatus comprising a buffer, the non-transitory computer-readable storage medium having recorded thereon:

audio video (AV) data;

a markup document to be preloaded into the buffer of the apparatus to enable the apparatus to reproduce the AV data in an interactive mode selected by a user of the apparatus, wherein the markup document does not comprise the AV data or any other AV data; and

control information providing functionality to enable the apparatus to identify buffering state information of the markup document to be preloaded into the buffer of the apparatus, the buffering state information being used by the apparatus in reproducing the AV data in the interactive mode selected by the user;

~~wherein:~~ wherein the control information comprises an application program interface (API) that generates a report signal ~~used to identify to determine~~ a buffering state of the markup document, and generates a return value capable of having any of three values, the return value having a first value if the markup document has been successfully preloaded into the buffer, a second value if the markup document cannot be read due to an error, and a third value if the markup document is being read; and

the apparatus uses the API to generate the report signal; and

~~is used by the apparatus~~ uses the return value generated by the API to verify whether the markup document has been successfully preloaded into the buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read.

2. (Canceled)

3. (Currently amended) The non-transitory computer-readable storage medium of claim 1, wherein the API comprises an [obj].isCached(URL, resType) API that generates the report signal, signal and the return value;

~~where the URL is a parameter indicating a file path of the markup document, document;~~
and

the resType is a parameter indicating an attribute of the markup document.

4. (Canceled)

5. (Currently amended) The non-transitory computer-readable storage medium of claim 1, wherein the control information further comprises an API that generates a fetch signal ~~used to issue in response to~~ a command to preload the markup document.

6. (Currently amended) The non-transitory computer-readable storage medium of claim 5, wherein the API that generates the fetch signal ~~returns a response~~ generates a return value indicating whether the command to preload the markup document has been successfully executed. ~~transmitted using the fetch signal.~~

7. (Previously presented) The non-transitory computer-readable storage medium of claim 1, wherein the control information further comprises an API that is used to determine whether preloading of the markup document is completed.

8.-9. (Canceled)

10. (Previously presented) The non-transitory computer-readable storage medium of claim 1, wherein the interactive mode is a mode in which the AV data is displayed in a display window defined by the markup document;

the apparatus is selectively operable in the interactive mode in which the AV data is displayed in the display window defined by the markup document, and a non-interactive video mode in which the AV data is displayed in the same manner as AV data recorded on a standard DVD; and

the user of the apparatus selects between the interactive mode and the non-interactive video mode.

11. (Previously presented) The non-transitory computer-readable storage medium of claim 1, further having recorded thereon a startup markup document separate from the markup document to be preloaded into the buffer of the apparatus and comprising preloading instructions enabling the apparatus to preload the markup document into the buffer of the apparatus;

wherein the selection of the interactive mode by the user causes the apparatus to read the startup markup document from the non-transitory computer-readable storage medium and execute the preloading instructions to preload the markup document into the buffer of the apparatus.

12.–13. (Canceled)

14. (Currently amended) An apparatus for reproducing audio video (AV) data using a markup document in an interactive mode selected by a user of the apparatus, the apparatus comprising:

a buffer to buffer the markup document to enable the apparatus to reproduce the AV data in the interactive mode selected by the user; and

a buffer manager to manage the buffer to preload the markup document and output buffering state information of the buffer in response to a report signal, the buffering state information being used by the apparatus in reproducing the AV data in the interactive mode selected by the user;

~~wherein;~~ wherein the apparatus generates the report signal using an application program interface (API); ~~and~~

the buffer manager uses the API to generate a return value capable of having any of three values in response to the report signal, the return value having a first value if the markup document has been successfully preloaded into the buffer, a second value if the markup document cannot be read due to an error, and a third value if the markup document is being read; and

the apparatus uses ~~the report signal is used by the buffer manager return value~~ to verify whether the markup document has been successfully preloaded into the buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read.

15. (Currently amended) The apparatus of claim 14, further comprising a content decoder to interpret the markup document, ~~and generate the report signal using the API;~~ API, and use the return value to verify whether the markup document has been successfully preloaded into the buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read.

~~wherein the buffer manager informs the content decoder of the buffering state information of the buffer in response to the report signal.~~

16. (Canceled)

17. (Currently amended) The apparatus of claim 15, wherein the API comprises a file path of the markup document and an attribute of the markup document as parameters.

18. (Currently amended) The apparatus of claim 15, wherein the API comprises an [obj].isCached(URL, resType) API that generates the report ~~signal,~~ signal and the return value; ~~where the URL is a parameter indicating a file path of the markup document,~~ document; and

the resType is a parameter indicating an attribute of the markup document.

19. (Canceled)

20. (Previously presented) The apparatus of claim 14, further comprising a content decoder to interpret the markup document;

wherein the buffer manager deletes the markup document from the buffer in response to a discard signal output from the content decoder.

21. (Previously presented) The apparatus of claim 20, wherein the content decoder generates the discard signal using a discard API.

22. (Previously presented) The apparatus of claim 14, wherein the interactive mode is a mode in which the AV data is displayed in a display window defined by the markup document;

the apparatus is selectively operable in the interactive mode in which the AV data is displayed in the display window defined by the markup document, and a non-interactive video mode in which the AV data is displayed in the same manner as AV data recorded on a standard DVD; and

the user of the apparatus selects between the interactive mode and the non-interactive video mode.

23. (Currently amended) An apparatus for recording and/or reproducing audio video (AV) data using a markup document in an interactive mode selected by a user of the apparatus before the apparatus reproduces any of the AV data, the apparatus comprising:

an AV buffer to buffer the AV data;

an AV reproduction engine to decode the AV data;

an enhanced navigation (ENAV) buffer to preload the markup document before the apparatus reproduces any of the AV data to enable the apparatus to reproduce the AV data in the interactive mode selected by the user;

an ENAV engine to interpret the markup document, and identify buffering state information of the markup document in response to a report signal, the buffering state information being used by the apparatus in reproducing the AV data in the interactive mode selected by the user; and

an I/O manager to obtain the markup document;

~~wherein:~~ wherein the ENAV engine generates the report signal using an application program interface (API);

the ENAV engine uses the API to generate a return value capable of having any of three values in response to the report signal, the return value having a first value if the markup document has been successfully preloaded into the buffer, a second value if the markup document cannot be read due to an error, and a third value if the markup document is being read; and

~~the report signal is used by the ENAV engine~~ uses the return value to verify whether the markup document has been successfully preloaded into the ENAV buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read.

24. (Currently amended) The apparatus of claim 23, wherein the I/O manager uses a blocked I/O method to ~~obtain-preload~~ the markup document from a data storage medium, and uses an unblocked I/O method to obtain-preload the markup document from a network.

25. (Currently amended) A method of reproducing AV data in an interactive mode using a markup document, the method comprising:

buffering the markup document to preload the markup document;

generating a report signal used to identify a buffering state of the markup document using an application program interface (API);

generating a return value capable of having any of three values in response to the report signal using the API, the return value having a first value if the markup document has been successfully preloaded into the buffer, a second value if the markup document cannot be read due to an error, and a third value if the markup document is being read; and

~~using the report signal~~ return value to verify whether the markup document has been successfully preloaded, whether the markup document cannot be read due to an error, and whether the markup document is being read; ~~read; and~~

~~outputting buffering state information of the markup document in response to the report signal.~~

26.–27. (Canceled)

28. (Currently amended) The method of claim 25, wherein the API comprises a file path of the markup document and an attribute of the markup document as parameters.

29. (Currently amended) The method of claim 25, wherein the API comprises an [obj].isCached(URL, resType) API that generates the report ~~signal~~, signal;
where the URL is a parameter indicating a file path of the markup ~~document~~, document;
and
the resType is a parameter indicating an attribute of the markup document.

30. (Currently amended) The method of claim 25, wherein the ~~outputting of the buffering state information comprises returning~~ return value has a value of 0 in response to if the markup document ~~being~~ has been successfully preloaded, ~~returning a value of 1 in response to if~~ the markup document ~~not being successfully preloaded~~ cannot be read due to an error, and ~~returning a value of 2 in response to if~~ the markup document ~~still is being preloaded~~ read.

31. (Previously presented) The method of claim 25, further comprising reproducing the AV data in the interactive mode using the preloaded markup document.

32. (Currently amended) A method of managing a markup document for use in reproducing AV data in an interactive mode, the method comprising:
buffering the markup document to preload the markup document in response to a fetch signal;
~~outputting~~ generating a report signal used to identify a buffering state of the markup document; in response to a report signal;
generating a return value capable of having any of three values in response to the report signal, the return value having a first value if the markup document has been successfully preloaded, a second value if the markup document cannot be read due to an error, and a third value if the markup document is being read;
~~staging~~ retrieving the preloaded markup document for decoding in response to a retrieve signal; and
~~deleting~~ indicating that the preloaded markup document is no longer needed in response to a discard signal.

33. (Currently amended) The method of claim 32, wherein the fetch signal is generated in response to a preload command; and
the method further comprising issuing a response comprises generating a return value indicating whether a the preload command to preload the markup document included in that caused the fetch signal to be generated has been successfully transmitted executed.

34. (Canceled)

35. (Currently amended) A method of managing a markup document for use in reproducing AV data in an interactive mode, the method comprising:
generating a fetch signal in response to a command to preload the markup document;
generating a report signal to determine a buffering state of the markup document;
generating a return value capable of having any of three values in response to the report signal, the return value having a first value if the markup document has been successfully preloaded, a second value if the markup document cannot be read due to an error, and a third value if the markup document is being read;
generating a retrieve signal to ~~stage retrieve~~ the preloaded markup document for decoding; and
generating a discard signal to ~~delete~~ indicate that the preloaded markup document is no longer needed.

36. (Currently amended) The method of claim 35, further comprising generating a release signal ~~in response to to delete~~ the preloaded markup document indicated as being no longer needed. ~~being presented.~~

37.-38. (Canceled)

39. (New) The non-transitory computer-readable storage medium of claim 1, wherein the error is an error caused by a physical error of the non-transitory computer-readable storage medium.

40. (New) The apparatus of claim 14, wherein if the markup document is being preloaded from a data storage medium, the error is an error caused by a physical defect of the data storage medium; and

if the markup document is being preloaded from a network, the error is caused by a connection disruption of the network.

41. (New) The apparatus of claim 23, wherein if the I/O manager is obtaining the markup document from a data storage medium, the error is an error caused by a physical defect of the data storage medium; and

if the I/O manager is obtaining the markup document from a network, the error is caused by a connection disruption of the network.

42. (New) The method of claim 25, wherein if the markup document is being preloaded from a data storage medium, the error is an error caused by a physical defect of the data storage medium; and

if the markup document is being preloaded from a network, the error is caused by a connection disruption of the network.

43. (New) The method of claim 32, wherein if the markup document is being preloaded from a data storage medium, the error is an error caused by a physical defect of the data storage medium; and

if the markup document is being preloaded from a network, the error is caused by a connection disruption of the network.

44. (New) The method of claim 35, wherein if the markup document is being preloaded from a data storage medium, the error is an error caused by a physical defect of the data storage medium; and

if the markup document is being preloaded from a network, the error is caused by a connection disruption of the network.